IPA Software Engineering Center
Information-technology Promotion Agency, Japan

IPAブース内セミナー　（ET2012）2012年11月14,15,16日

組込みソフトウェアテスト事例集の紹介
組込みソフトウェア開発における品質向上の勧め
[テスト編～事例集～]

独立行政法人 情報処理推進機構（IPA）
技術本部 ソフトウェア・エンジニアリング・センター(SEC)

日本電気通信システム　羽田　裕

Copyright© 2012 Information-technology Promotion Agency, Japan. All rights reserved.

組込みソフトウェアテスト事例集

Practical Testing Examples

SEC
Software Engineering for Mo·No·Zu·Ku·Ri

新刊

Recommendations for Improvement of Quality of Embedded Software Development [w/ Practical Testing Examples]

2012年11月12日発行

書籍：

A5判、100ページ

定価500円（税込）

pdf版：

http://sec.ipa.go.jp/publish/index.html#emb

無償

Copyright © 2012 IPA, All Rights Reserved.

目次

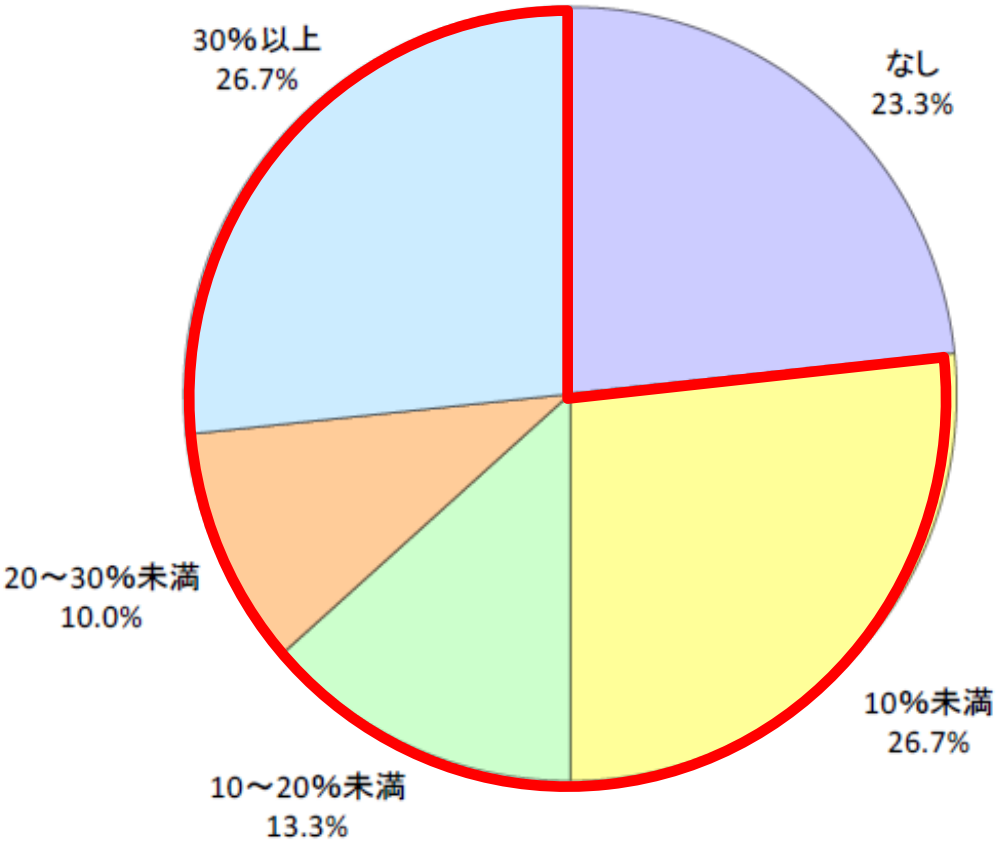
1. テストの重要性と課題
   1. 組込みシステムの不具合発生状況(産業実態把握調査)
   2. 組込みソフトウェアのテスト実態調査結果
2. テスト事例集のコンセプト
3. テスト事例集の構成
4. 利用方法
5. 事例紹介

Copyright © 2012 IPA, All Rights Reserved.

1.1 組込みシステムの不具合発生状況-1

2012年版組込みソフトウェア産業実態把握調査報告書

不具合発生製品率



75%以上の製品で出荷後に不具合発生

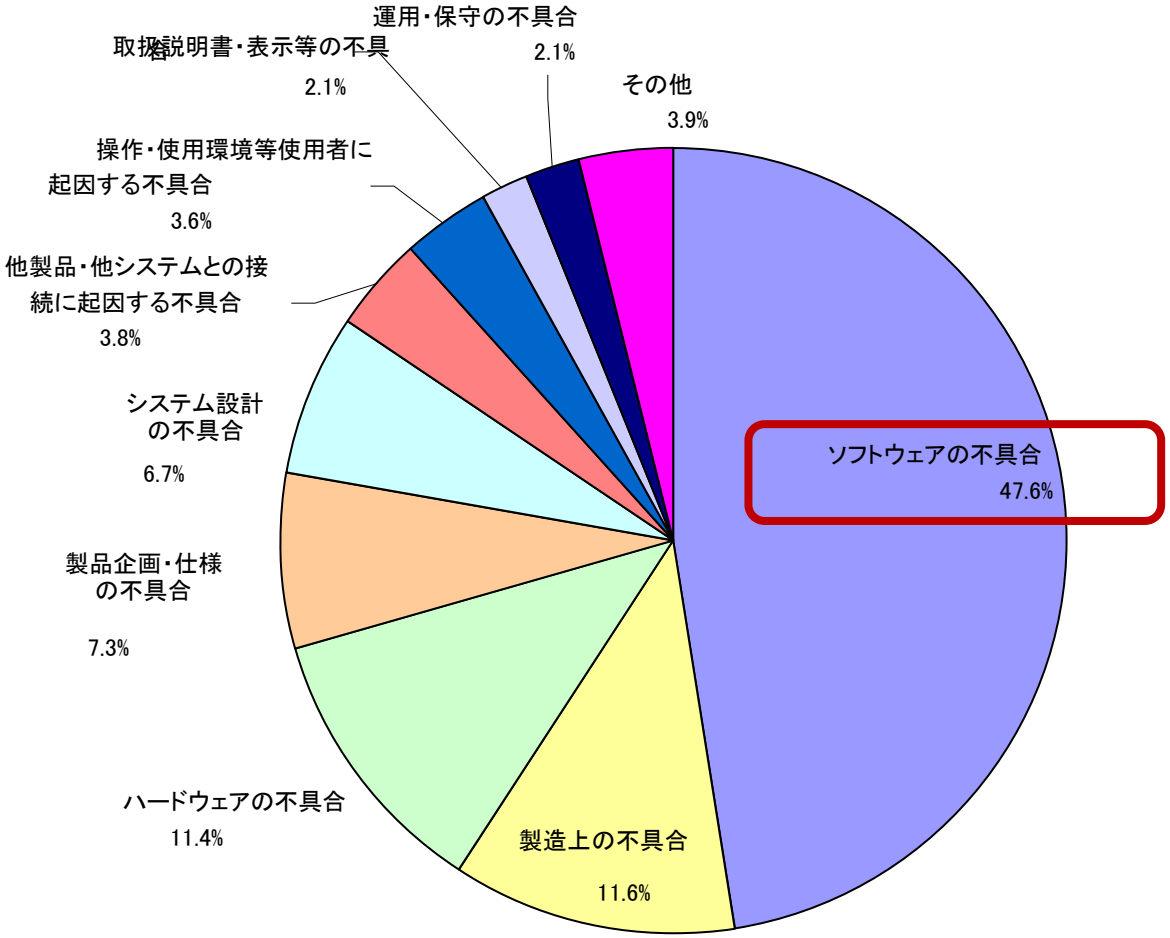
Copyright © 2012 IPA, All Rights Reserved.

1.1 組込みシステムの不具合発生状況-4

SEC
Software Engineering for Mo·No·Zu·Ku·Ri

2011年版組込みソフトウェア産業実態把握調査報告書

出荷後の不具合原因(件数ベース)



不具合原因で最も多いのがソフトウェア不具合

Copyright © 2012 IPA, All Rights Reserved.

課題（アンケート、ヒアリングの結果）

必要だが、無いと考えているもの

① ・テスト十分性を評価する尺度

1．現実的な品質とコストのバランスのとり方
2．開発現場の事情・状況に即した、
3．『実用的』な標準、基準とその策定の考え方
4．手法の『実用的』な活用方法、勘所

⑤ ・テスト工程毎の目的・範囲の標準

 Copyright © 2012 IPA, All Rights Reserved.

IPAへの期待（アンケート、ヒアリングの結果）

標準や基準と、その基本となる考え方が欲しい

汎用的な解はなさそう。
理論的な解はあっても、実用的な解はない。
　⇒事例を参考にしたい。

自社の事例は出せないが、他社の事例は欲しい

自社のテスト基準や考え方を開示すると、
ネガティブキャンペーンに使われかねない
⇒他社とベンチマークして自社の妥当性を評価したい

Copyright © 2012 IPA, All Rights Reserved.

目次

1. テストの重要性と課題
2. **テスト事例集のコンセプト**
   1. **前提**
   2. **ESxRシリーズにおける位置付け**
3. テスト事例集の構成
4. 利用方法
5. 事例紹介

Copyright © 2012 IPA, All Rights Reserved.

2．1　前提

- いくらテストしても、バグ0にはならない
- テスト偏重による弊害がある
- 品質とコストのバランスが必要

この前提のもとに、
先進企業の成功･効果事例を収集し、
公開可能な表現に加工し、
事例集として発行する。

組込み開発リファレンスという位置付けのESTR（*Embedded System development Testing Reference*）とはせず、その前段の事例集とした。

Copyright © 2012 IPA, All Rights Reserved.

2．2　ESxRシリーズにおける位置付け

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

For Embedded
Software
Development

組込みソフトウェア開発技術リファレンス

ESxR

Embedded System development exemplar Reference

Series

品質の高いソフトウェアを作るにはどのような作業をどのように進めたらよいのでしょうか？ソフトウェア開発の作業や品質を見えるようにするにはどのようにしたらよいのでしょうか？ESxRシリーズは、こんな疑問に答えるためのソフトウェア開発に関する標準リファレンスです。

IPA Software Engineering Center
Information-technology Promotion Agency, Japan

ESxR Embedded System development exemplar Reference

プログラムへ品質を作り込む

ESCR P4
Embedded System development
Coding Reference guide
組込みソフトウェア実装時にソースコードの品質を向上させるためのコーディングルールなどを作法として整理しました

開発プロセスを整備する

ESPR P6
Embedded System development
Process Reference guide
高品質なソフトウェアを開発するための作業や手順を、標準的な開発プロセスとして整理しました

プログラム計画書を洗練する

ESMR P8
Embedded System development
Management Reference guide
チームやプロジェクトでソフトウェアを開発する際には不可欠な開発計画書の記述内容や記述フォームを整理しました

品質を計測しコントロールする

ESQR P10
Embedded System development
Quality Reference guide
ソフトウェアの品質や開発作業の質を見える化し、客観的に評価改善するための品質定量化の指標や考え方を整理しました

現場のノウハウを設計に活かす

ESDR P12
Embedded System development
Design Reference guide
現場の問題意識を反映した事例をもとに、様々な設計場面でのノウハウを経験の浅い設計者でも利用しやすい形に整理し提示しました

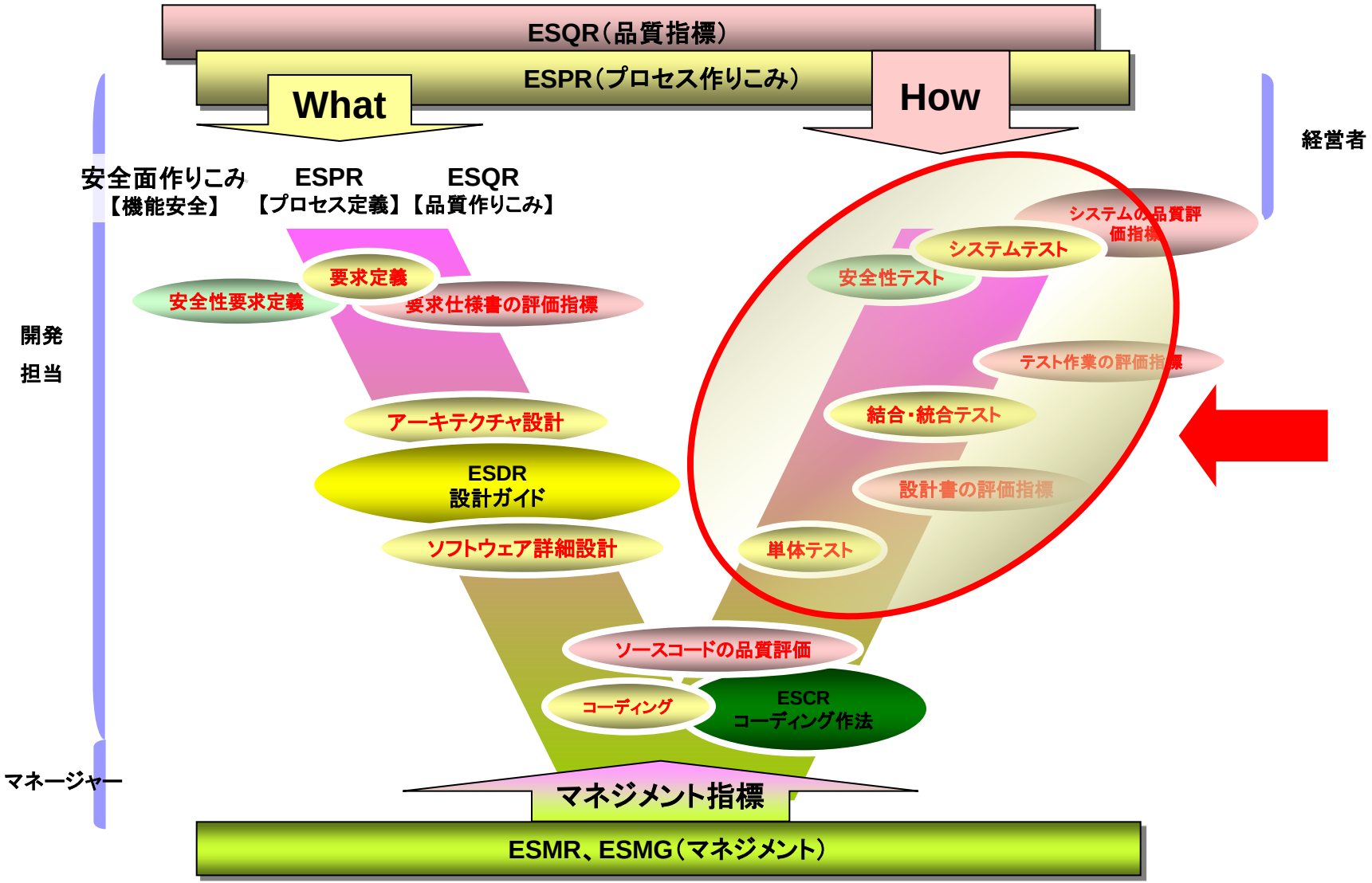
V&V改善を実践する

ESTR P14
Embedded System development
Testing Reference guide
リファレンスとしてのESTR策定に向けて、先進企業におけるV&V改善の実践事例を読みやすくまとめました

Copyright © 2012 IPA, All Rights Reserved.

2．2　ESxRシリーズにおける位置付け

SEC
Software Engineering
for Mo･No･Zu･Ku･Ri



Copyright © 2012 IPA, All Rights Reserved.

目次

1. テストの重要性と課題
2. テスト事例集のコンセプト
3. **テスト事例集の構成**
    1. **3つのPartで36事例を解説**
    2. **各項の構成**
4. 利用方法
5. 事例紹介

Copyright © 2012 IPA, All Rights Reserved.

3．1　3つのPertで36事例を解説

SEC
Software Engineering for Mo·No·Zu·Ku·Ri

【事例構成】3つの視点で36事例を分類

Part 1.　テストの役割と限界　　19種

- テストの役割　（テストの実施方法、考え方・・・）
- テストの限界　（効率的なテスト方法・・・）
- 基準値·目標値·終了条件　（尺度等の標準化・・・）

Part 2.　テストへの要求と対応状況　　8種

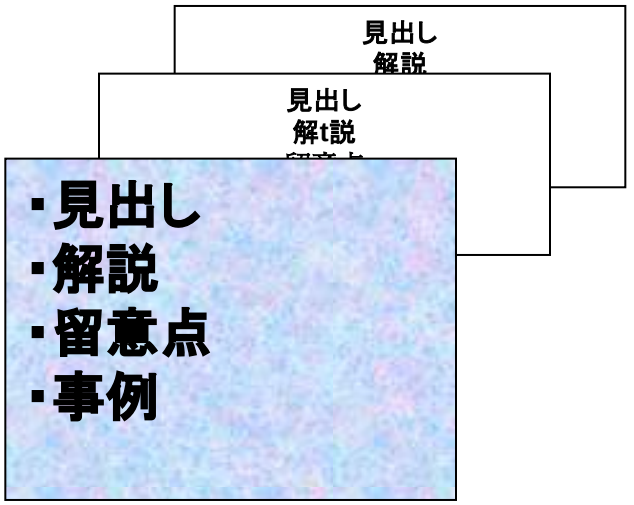
- ツール活用　（効率化のためのツール活用・・・）
- 公的機関における基準　（公的機関基準等の活用）

Part 3.　テストの基本的テクニック　　9種

- テスト技術·技法の分類　（項目十分性の追及・・・）
- 環境　（性能測定環境不可・・・）
- 教育　（ユーザ視点教育・・・）

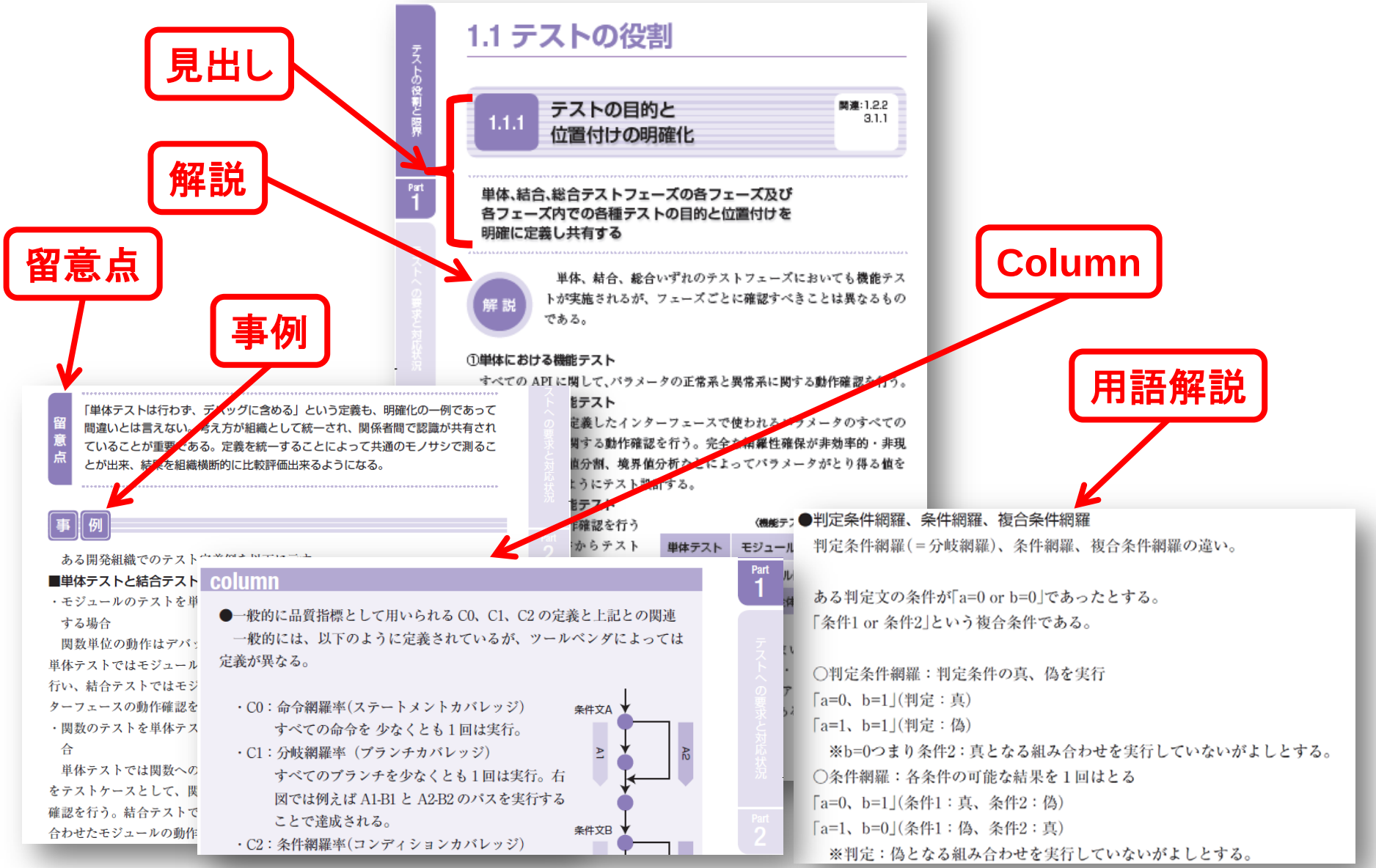
【事例記述構成】

個別課題と対策事例を整理



Copyright © 2012 IPA, All Rights Reserved.

3．2　各項の構成



Copyright © 2012 IPA, All Rights Reserved.

3．2　各項の構成

SEC
Software Engineering
for Mo･No･Zu･Ku･Ri

見出し	各事例での対策における基本的な考え方、あるいは重要事項を端的に記述
解説	開発現場で実践されている事例の内容を図表などを用いて解説
留意点	留意すべき点、関連するトピックを記述
事例	実際の具体例をポイントを示しながら、公開可能な表現で記述
Column	事例解説に関連した事項を記述
用語解説	本書で頻繁に用いる用語や勘違いしやすい用語を解説

Copyright © 2012 IPA, All Rights Reserved.

目次

1. テストの重要性と課題
2. テスト事例集のコンセプト
3. テスト事例集の構成
4. **利用方法**
5. 事例紹介

Copyright © 2012 IPA, All Rights Reserved.

4　利用方法

次のような場合に利用していただくことを想定しました

- 論理的に妥当性を証明し難い「当たり前」の裏付け
- 自社の場合と比較して、自社の取り組みを評価
- 自社の基準・標準を設定する際の参考
- 自社の基準・標準の妥当性を議論するきっかけ

事例に対して異論もあるかと思います。
「派生開発には使えない」、「PJ毎に事情が異なる」・・・

大いに議論していただけるきっかけになれば幸いです。
また、事例は事実に基づいています。

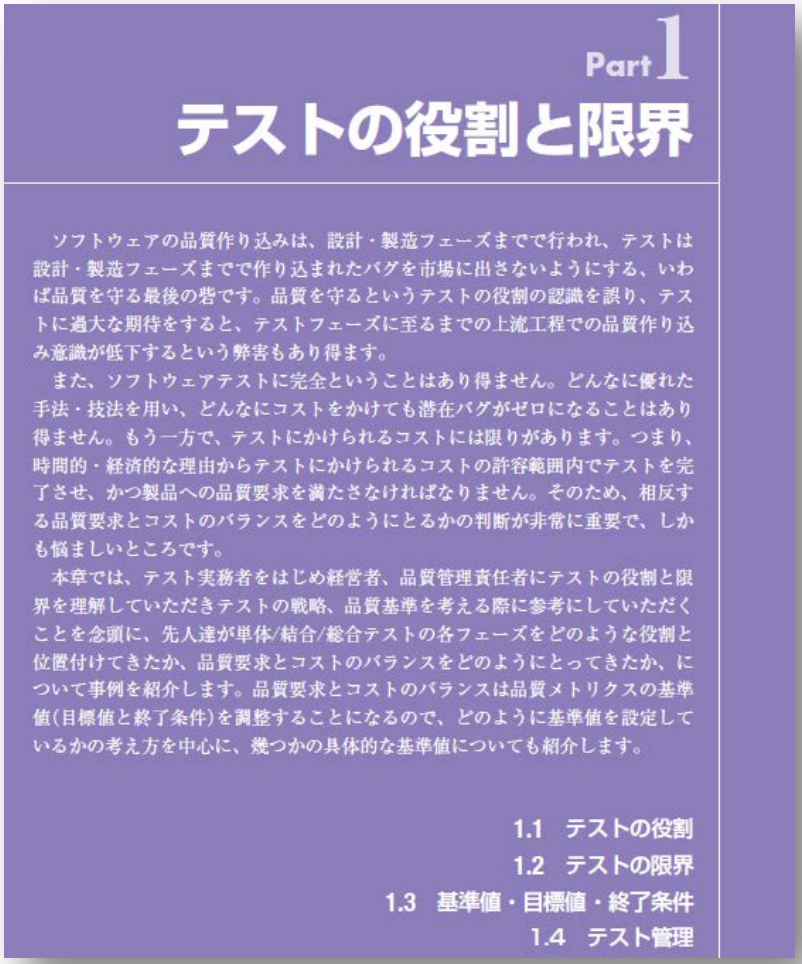
Copyright © 2012 IPA, All Rights Reserved.

目次

1. テストの重要性と課題
2. テスト事例集のコンセプト
3. テスト事例集の構成
4. 利用方法
5. **事例紹介**
    1. **Part 1　テストの役割と限界**
    2. **Part 2　テストへの要求と対応状況**
    3. **Part 3　テストの基本的テクニック**

Copyright © 2012 IPA, All Rights Reserved.

Part 1　テストの役割と限界



Part 1

テストの役割と限界

ソフトウェアの品質作り込みは、設計・製造フェーズまでで行われ、テストは設計・製造フェーズまでで作り込まれたバグを市場に出さないようにする、いわば品質を守る最後の砦です。品質を守るというテストの役割の認識を誤り、テストに過大な期待をすると、テストフェーズに至るまでの上流工程での品質作り込み意識が低下するという弊害もあり得ます。

また、ソフトウェアテストに完全ということはあり得ません。どんなに優れた手法・技法を用い、どんなにコストをかけても潜在バグがゼロになることはあり得ません。もう一方で、テストにかけられるコストには限りがあります。つまり、時間的・経済的な理由からテストにかけられるコストの許容範囲内でテストを完了させ、かつ製品への品質要求を満たさなければなりません。そのため、相反する品質要求とコストのバランスをどのようにとるかの判断が非常に重要で、しかも悩ましいところです。

本章では、テスト実務者をはじめ経営者、品質管理責任者にテストの役割と限界を理解していただきテストの戦略、品質基準を考える際に参考にしていただくことを念頭に、先人達が単体/結合/総合テストの各フェーズをどのような役割と位置付けてきたか、品質要求とコストのバランスをどのようにとってきたか、について事例を紹介します。品質要求とコストのバランスは品質メトリクスの基準値(目標値と終了条件)を調整することになるので、どのように基準値を設定しているかの考え方を中心に、幾つかの具体的な基準値についても紹介します。

1.1　テストの役割
1.2　テストの限界
1.3　基準値・目標値・終了条件
1.4　テスト管理

テストは最後の砦

テスト偏重による弊害

限界：バグ0にはならない

限界：テストボリュームの爆発

限界：かけられるコストの制約

品質要求とコストのバランス

Copyright © 2012 IPA, All Rights Reserved.

事例紹介　1.1.3

1.1.3 模擬環境と実機環境の相互補完

関連:2.1.7
3.2.1

模擬環境でテストしても、実機環境でのテストも不可欠である。
模擬環境と実機環境で実施するテスト項目の
切り分けを明確にして、漏れと重

解説

模擬環境は実機環境を完全に
境と実機環境では、できること/で
と/実行が困難なことが異なり、
そのように模擬環境を構築す

シミュレータなど模擬環境によるテスト
るが、事前に想定した動作・現象を再現さ
ドウェア設定・制御やリソース/タイミン
難である。こうした模擬環境でテストで
テスト実施が不可欠である。

一方、実機環境でも、テストに必要なす
ではないので、範囲と理由を明確にして実機環
る。

どちらの環境でもテスト実施が可能ならば、どちら
を考えてテスト設計する。

シミュレーション　　実機テスト

留意点

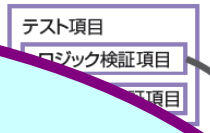
◆模擬環境でテストする場合は、可能であれば、どのテスト項目が実機と同等であるかの根拠を明確にしておくことがアカウンタビリティ確保のために有用である。
◆例えば、システムシミュレータの中には、100%実機と同じソースコードを使い、非常に高い精度（命令レベルで 98%など）で実機と同じタイミングを再現しているものもある。その場合には「98%実機と同等とみなせる」とするなど。

事例

ある製品開発における事例を以下に示す。

■ハードウェア開発と並行してソフトウェア開発を実施するために、ハードウェア仕様書に記載された入出力インターフェースを忠実にモデル化した模擬環境を構築し、そこで製品に搭載するデバイスドライバや上位層のアプリケーションソフトウェアなどのテストを実施した。

■実機のハードウェア仕様のインター



以下のことを明確に

・何の目的でそのシミュレータを用いるのか
・そのためには、どこまでの動作が実機と同じである必要があるのか。つまり、どこから先をモデル化してどの程度抽象化するのか
これらを明確にしてシミュレーション環境を構築し、テスト項目を抽出すること。この判断/明確化を誤ると、
・構築においては、無駄なシミュレーション環境構築作業をした挙句、遅くて使い物にならないシミュレータを構築してしまう。ハードウェア検証用シミュレータをそのままソフトウェア検証用に用いて失敗するのもこのケースである。
・テスト項目抽出においては、シミュレータ上で意味の無いテストを実施して、実機テストとの重複実行をしてしまう。

模擬と実機それぞれのテストの位置付けや、適用に際しての根拠などが組織・部門で明確になっていますか？

Copyright © 2012 IPA, All R

1.1.3に関連する用語解説

■模擬環境

一般的に開発現場で用いられる模擬環境には、以下の種類がある。

●シミュレーション(Simulation)

一般用語：ふりをする(外見を同じようにする)。模擬実験。

コンピュータ業界用語：インターフェースを再現する。

●エミュレーション(Emulation)

一般用語：見習う(中身を同じようにする)。模倣。

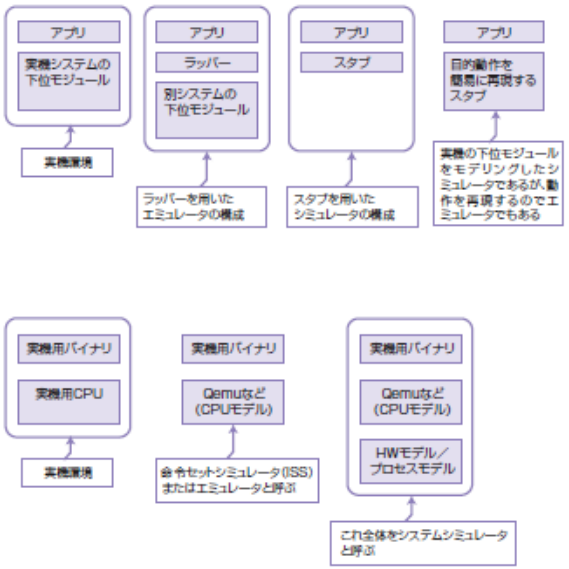
コンピュータ業界用語：中身の動作までを再現する。

●ラッパー(Wrapper)

ラッパーは、異なるシステムで元のシステムの動作を代用するためにインターフェース調整をするモジュールである。つまり中身の動作を再現するので、エミュレータと言える。一方で、ラッパーはインターフェースを再現するのでシミュレータとも言える。

●スタブ(Stub)

スタブは、インターフェースを再現するのでシミュレータである。しかし、動作を再現するスタブもあり、その場合はエミュレータと言える。



Copyright © 2012 IPA, All Rights Reserved.

事例紹介　1.3.7

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

1.3.7 達成不可能なコードカバレッジ基準とその終了可否判断

関連：1.2.4

コードカバレッジ基準に満たない場合は
その理由を明確にして合理的に判断する。

解説

単体テストの結果、設
満たなかった場合、その結
なることがある。また、基準値
いというような現実的な状況に
また、差分開発・流用開発においては、
が存在することまで把握できないことも
このような場合、網羅しきれなかった
に応じて別の指標と組み合わせて評価
ト終了とするのが一般的である。

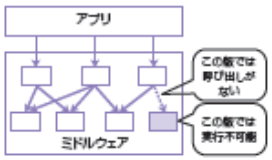
留意点

絶対に通過しないはずのパスでも、実際
る。例えば、下記事例のように上位関数で
上位関数でのチェック方法に誤りがあったり、
クを削除したときなど。

事例

ある組込み製品を開発している組織で、網羅率100%を目標としたが、100%を達成できずに、無駄な時間・工数を費やした事例を以下に示す。

■汎用性を考慮したミドルウェアの中の内部関数に、そのバージョンのシステムでは使われない関数が含まれていた。構造解析ツールで調べても、使用される関数からの呼び出し関係があり、実際には当該内部関数が使用されないこと



が自明ではなかったため、通過し得ないことが判明するのに時間がかかった。

■入力パラメータのチェックを必須とし、更に if then else の対応付けを行うようコーディング規約に定められていたため機械的に else 文を挿入したが、上位関数でチェック済みであったため、実行不可能なパスとなり、カバレッジ100%になり得ないケースが生じた。

行されるようにテストを設

nch Coverage：

場合がそれ

真偽が少な

個々の条件の真

現するようにテスト

erage：MCC)

すべての条件で、あり得るすべての結果の組み合せが少なくとも1回は出現するようにテストを設計する 。

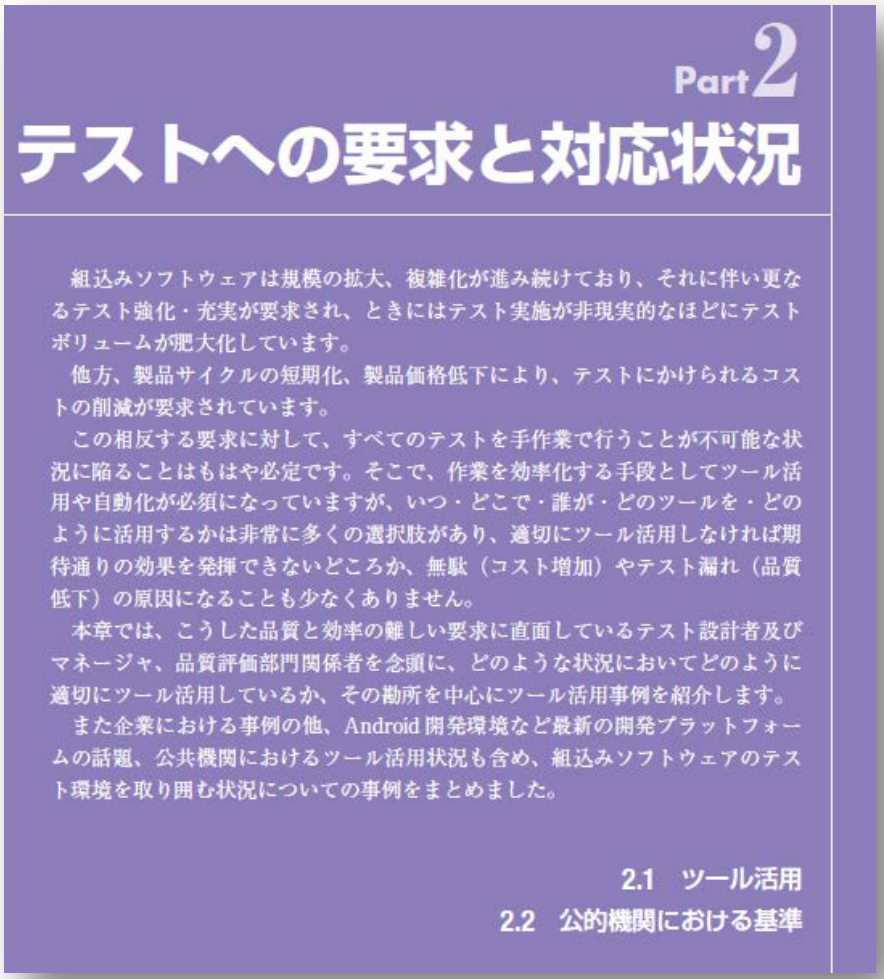
・経路組み合わせ網羅(path coverage)

コード内のすべてのパス(制御パス)が少なくとも1回は実行されるようにテストを設計する。

「C0 100%は必須」が組織の暗黙の不文律になっていたり、トップ指示になっていて、苦慮しているということはありませんか？

Copyright © 2012 IPA, All Rights Reserved.

Part 2　テストへの要求と対応状況



Part 2

テストへの要求と対応状況

組込みソフトウェアは規模の拡大、複雑化が進み続けており、それに伴い更なるテスト強化・充実が要求され、ときにはテスト実施が非現実的なほどにテストボリュームが肥大化しています。

他方、製品サイクルの短期化、製品価格低下により、テストにかけられるコストの削減が要求されています。

この相反する要求に対して、すべてのテストを手作業で行うことが不可能な状況に陥ることはもはや必定です。そこで、作業を効率化する手段としてツール活用や自動化が必須になっていますが、いつ・どこで・誰が・どのツールを・どのように活用するかは非常に多くの選択肢があり、適切にツール活用しなければ期待通りの効果を発揮できないどころか、無駄（コスト増加）やテスト漏れ（品質低下）の原因になることも少なくありません。

本章では、こうした品質と効率の難しい要求に直面しているテスト設計者及びマネージャ、品質評価部門関係者を念頭に、どのような状況においてどのように適切にツール活用しているか、その勘所を中心にツール活用事例を紹介します。

また企業における事例の他、Android 開発環境など最新の開発プラットフォームの話題、公共機関におけるツール活用状況も含め、組込みソフトウェアのテスト環境を取り囲む状況についての事例をまとめました。

2.1　ツール活用
2.2　公的機関における基準

テストボリューム肥大化

テスト期間短縮要求

ツールによる自動化・効率化

ツール導入時の課題

Android開発のテスト環境

Copyright © 2012 IPA, All Rights Reserved.

事例紹介　2.1.3

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

2.1.3 ツールのポータル化

関連：2.1.1/
2.1.2/2.1.4/
2.1.5/2.1.6

ツールを活用するには、導入後のメンテナンスコスト負担の問題を解消する必要がある。とくに大きな組織でツールを統一する場合には、ツールをポータル化してサービス提供することも選択肢に入れる

解説

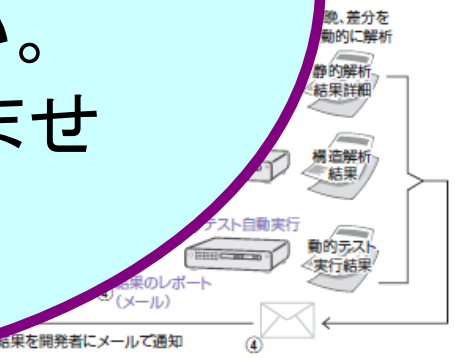
ツールを適用す
グループ会社全体な
ツールを適用するには
①ツール利用環境構築
②利用者への運用教育
③ライセンス費用
④保守サービス費用
などのイニシャル/ランニング・コス
教育が必要になるが、全社やグルー
適用する際には、全利用者にそれら
うな場合には、推奨ツールをポータル
負担を軽減する方法がある。
ポータル化によって、導入・運用常態化

留意点

◆ツール利用のタイミングが遅れることがないよう
ルを使えるようにサービス提供すべきである。
◆静的解析ツールの利用タイミングが遅れたため、ツールの指摘に
(=開発者)の対応が表面的なものになってしまい、ツール適用効果が得られなかったという事例がある。

事例

ツール利用のタイミングについて、以下に例を示す。
ソースコード静的解析ツールとソースコードバージョン管理ツールとを連携し、夜間バッチを利用して、チェックインされたソースコードを自動的に静的解析ツールにかけて、朝には担当者に結果がメールで配信される仕組みを構築した。
①開発者は、帰宅前にソースコードをリポジトリにチェックインする
インテグレーションツールがソースコードの差分を読み
静的解析、構造解析、ビルド、更
行する
結果を朝までに開発者宛
前提となってい



ツール活用の効果は分かってはいるけれど、このPJで、そんな高価なツールを買う余裕はない。そんなケースがありませんか？

Copyright © 2012 IPA, All Rights Reserved.

事例紹介　2.1.7

2.1.7 Android開発での模擬環境と実機環境の使い分け

関連: 1.1.3, 3.2.1

シミュレーション環境と実機環境でのテストケースを切り分ける。

解説

Androidには豊富な開発環境が整備されている。テスト環境としては、OSS(オープン・ソース・ソフトウェア)で高速なバイナリシミュレータQemu(キューエミュ)を用いた仮想HW(goldfish)が用意されており、PC上で製品としてのシステムシミュレーションが可能になっている。

goldfishは、基本HWのモデルをそろえている。その他の周辺HWについては、利用者がモデルを作成する。そのためのツールとして、Android仮想デバイス(Android Virtual Device : AVD)の設定をサポートしている。

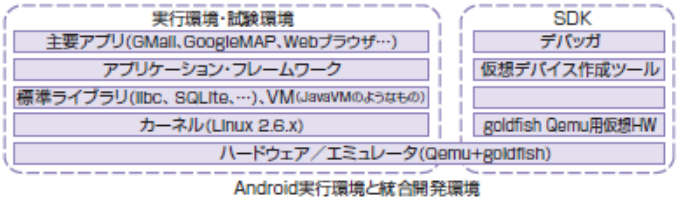
ただし、goldfishによるシミュレーション環境は、実機用のソフトウェアをそのまま使用出来るので機能面はかなり忠実に(低レベルの抽象度で)模擬するが、タイミングについては実機のタイミングを模擬していない。

つまり、機能テストについては、製品を構成する部品がそろった環境で実機用のカーネルから、デバイスドライバ、ミドルウェア、アプリまでを統合してテスト出来るので、ほとんどのテストケースを実行することが出来る。とは言っても、製品の外部については適切なモデルを作成して組み込む必要があり、それはたやすいことでないことに注意すべきである。

一方、タイミングに関するテストは、実機での実行結果と異なるため、タイミングに依存するテストケースは実機環境で行わなければならない。実機環境で実施するためのテストプログラムの検証にシミュレーション環境を使用することによって、実機環境使用時間を短縮出来る。

シミュレーション環境で出来るテストケース、実機で実施すべきテストケース、実機使用前にシミュレーション環境で準備出来る作業を事前に切り分けて、以下のことを心がけるべきである。

①無意味なテストを実施しない(不適切なテスト環境でのテスト実施排除)。
②同じテストケースの重複テスト実行を避ける(無駄を排除)。
③実機環境での作業時間を最小にする(テスト効率化)。



Android実行環境と統合開発環境

留意点

製品の外部については適切な抽象度のモデル(外界モデル)を作成して組み込む必要があり、それはたやすいことではない。テスト対象が何であるかを明確にした上で、作成するモデルの抽象度を決定する。いたずらに抽象度を下げて現実の動作を、仕組みまで含めて忠実にモデリングすることが優れたモデリング技術ではないことを理解すべきである。

事例

Qemuを用いたシステムシミュレーションは、タイミングを模擬出来ないという短所があるが、Qemuにアドオンする形で実機と同じタイミングを再現する方法について、以下に示す。

■論文「QEMUによるHW/SW 協調シミュレータの構築」(設計・検証技術、一般セッション、アーキテクチャ、情報処理学会創立50周年記念)の方法によると、ソフトウェア内部のタイミング(命令レベルの時間精度)を保証し、かつ周辺ハードウェアモデルに設計通りのタイミングを再現させることが出来るため、タイミングに依存するテストケースについても、かなりの部分を実機レスでテストすることが出来る。この方法は実機と同等の実行性能であることも大きな特徴である。

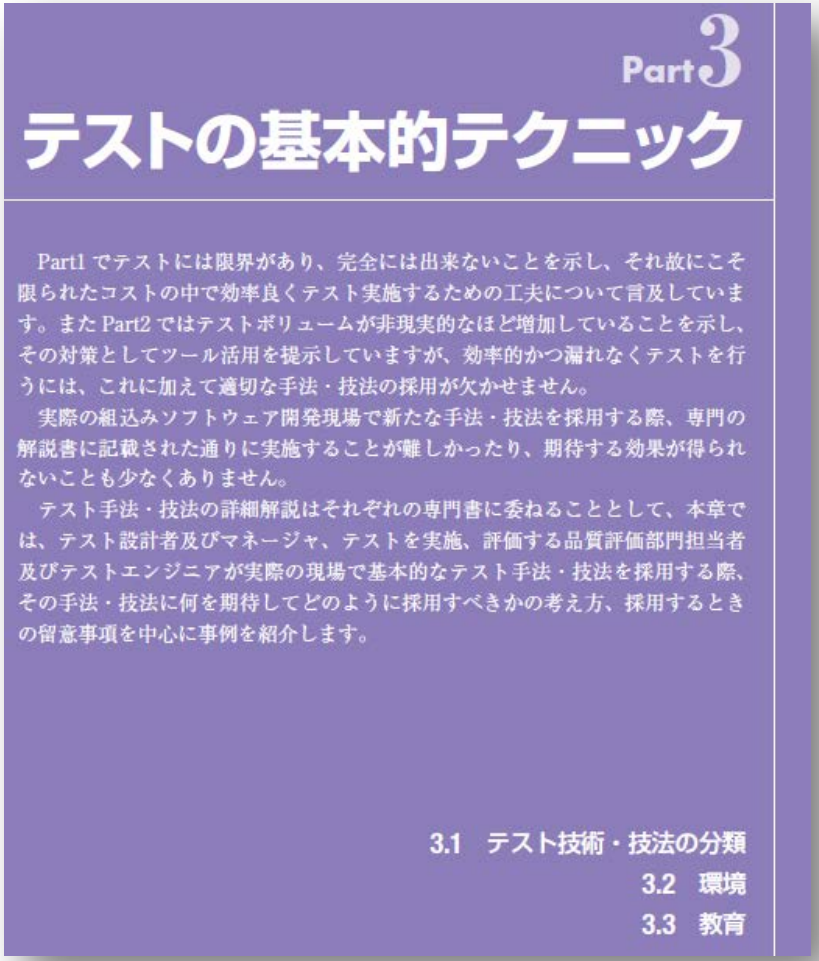
■単純にタイミングを再現するだけであれば、それほど高度な技術を必要としないが、実行性能が実機の何百倍も遅かったら、ソフトウェア検証用として用をなさない。ハードウェア検証用シミュレータには、この手のものが多いので、シミュレータを選択するときには、ハードウェア検証用シミュレータとソフトウェア検証用シミュレータの違いを意識する必要がある。前述の論文では、高速なソフトウェア検証用シミュレータとしてQemuを用いた。

column

CPU動作をモデル化することによって、"コンピュータを使ってコンピュータをシミュレートする"ものを、エミュレータあるいは命令セットシミュレータ(ISS : Instruction Set Simulator)と呼ぶ。Qemuはこれに属する。

一般的にISSでは高速化のため、1命令ごとにシミュレートするのではなく、命令ブロック(ベーシックブロックとも呼ぶ)単位でシミュレートする。

Copyright © 2012 IPA, All Rights Reserved.



適切な手法･技法の採用

手法･技法採用時の留意事項

テストエンジニア育成

Copyright © 2012 IPA, All Rights Reserved.

IPAセミナー

SEC
Software Engineering
for Mo·No·Zu·Ku·Ri

2012年11月16日（金）@会議センター3階【303】

第7部 13:30～14:30

新　刊

- 組込みソフトウェア向け設計ガイド ESDR[事例編]紹介
- IPA/SEC研究員　浜田直樹、三菱電機メカトロニクスソフトウエア（株）　岩橋正実、横河医療ソリューションズ（株）　楢木野公彦

第8部 14:45～15:45

新　刊

- 組込みソフトウェアテスト事例集の紹介
- IPA/SEC研究員　石井正悟、石田茂

第9部 16:00～16:45

予　告

- 組込みソフトウェア開発におけるバグ管理ガイドの紹介
- 日本電気（株）　三橋二彩子

Copyright © 2012 IPA, All Rights Reserved.

IPAブースプレゼン

SEC
Software Engineering for Mo·No·Zu·Ku·Ri

ESDR

- 11月16日10:30～10:50
- IPA/SEC　浜田直樹

テスト事例集

- 11月16日11:00～11:20
- IPA/SEC　石井正悟、富士通コンピュータテクノロジーズ　馬場匡史、日本電気通信システム　羽田裕

ESxR概要

- 11月16日13:00～13:20
- IPA/SEC　石田茂、村松昭男

トレーナーズトレーニング

- 11月16日12:30～12:50
- IPA/SEC　松田充弘、石井正悟

Copyright © 2012 IPA, All Rights Reserved.

ご清聴ありがとうございました

Copyright © 2012 IPA, All Rights Reserved.